Remarks

Office Action of February 2, 2007

1. Rejections Under 35 USC 102.

Claims 1, and 6 were again rejected under 35 USC 102(e) as being anticipated by US Published Patent Application 20020138617 to Christfort as follows

Claim No.	Christfort
1	¶0067, ¶0087, ¶0094, ¶0163, ¶¶0164-0165, ¶¶0167-0168
6	¶0067, ¶0087, ¶0094, ¶0163, ¶¶0164-0165, ¶¶0167-0168

2. Rejections Under 35 USC 103.

Claims 2, 3, 5, and 7 were again rejected under 35 USC 103(a) as being unpatentable over Christfort as applied to claims 1, 4, and 6 and further in view of US Patent 6,223,180 to Moore.

Claim No.	Christfort	Moore
2	¶0059	Col. 7/16-18, Col.2/64-65
3	¶0004, ¶0067	Fig. 2, item 26
4	¶0067, ¶0087, ¶0094, ¶0163, ¶¶0164-0165, ¶¶0166-0168	
5	¶0059	Col. 7/16-18, Col.2/64-65
7	¶0059	Col. 7/16-18, Col.2/64-65

Remarks/Arguments

With reference to the Final Action of July 10, 2006, restated in the Advisory Action of November 15, 2006, and the Action of February 2, 2007, Applicants offer the following remarks.

The Art of Record

United States Patent Application 20020138617 to Christfort et al for Providing Content From Multiple Services describes a method and apparatus for providing a network based operating system for mobile clients. Christfort discloses that services may be developed that can be used to support different client devices with different capabilities. The services provide output with multiple variations based on different devices, and an intermediary selects the variation best suited for the requesting device. To be note is that Christfort asserts that an online software development system is provided to allow services to create, edit, test, and deploy applications at an intermediary using only a browser at the client end. Christfort et al. describe that services may also be provided that can be accessed and referred to by other services, facilitating the combining of different services. Services may also store and access data at an intermediary using variables and a mapping of the stored data to the variables. Data stored at the intermediary may be used to allow an end user to return to a previously accessed service.

Specific paragraphs of Christfort et al. cited in the Office Action include paragraph [0004], paragraph [0059][i], paragraph [0063][ii], paragraph [0067][iii], paragraph [0087][iv], paragraph [0091][v], paragraph [0094][vi], paragraph [0163][vii], paragraph [0164][viii], paragraph [0166][ix], paragraph [0167][x], and paragraph [0168][xi], (all paragraphs are collected in endnote at the end of this submission)

United States Patent 6,223,180 to Moore et al. for System And Computer Implemented Method For Transforming Existing Host Based Screen Applications Into Components Useful In Developing Integrated Business Centric Applications describes a system and method in a computer system having a repository for storing data. The method is

implemented by the computer system. The method encodes display, entry fields and static text of a screen application (screen data) into Host Reply Definition (HRD), Request (REQ) and recognition files. These are then stored in the repository. A graphical user interface program is used for building and transforming the HRD, REQ and files stored in the repository into components. Next, the HRD, REQ and recognition files are extracted from the repository and associated with the screen application. The attributes of these files are written into a type library, forming the software components. After this, the recognition file is stored in a directory structure independent of the repository. Finally, the components are registered in a registry for recognition by other applications and components.

Specific provisions of Moore et al. cited in the Final Action include Column 4, lines 62-65[xii], and Column 7, lines 16-18[xiii]. (Collected in the Endnotes at the end of the submission).

Applicants' Claimed Invention

Status of the Claims.

In the Office Action of February 2, 2007 all of the claims were rejected. The claims have been amended.

Exemplary Claim

Claim 1, as amended, is exemplary.

1. (Currently Amended) A method of rendering an object from a text and numeric centric line of business application to a graphical user interface centric content manager client application said text and numeric centric line of business application resident on a a resource manager, said resource manager including

(i) an HTTP server,

(ii) a store object agent,

(iii) a resource manager tracking table data base, and

(iv) a file system;

said method comprising:

a. requesting the object from a text and numeric centric line of business application;

b. the line of business application initiating an associated host initiated display application program interface, and calling a workstation listener;

c. a content manager host sending customer application request to a workstation listener;

d. the workstation listener launching an associated content manager graphical user interface client;

e. the content manager graphical user interface client building a request for the object and sending the request to the associated content manager application for the associated host initiated display; and

f. the content manager application responding to the graphical user interface centric content manager client and rendering the object from the text and numeric centric line of business application to the graphical user interface centric user.

Discussion

The Pending Claims, As Amended, Are Properly Allowable to Applicants Over The Art of Record

Applicants have previously amended the claims as suggested by the Examiner and to more particularly clarify and point out the invention. Specifically, the Final Action recites:

> "Examiner respectfully disagrees. In response to Applicant's arguments, the recitation of 'method of rendering an object from a text and numeric centric line of business application to a graphical user interface centric content manager client application' has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone." (Citations omitted).

Accordingly, applicants have previously amended the independent claims, as illustrated by claim 1.

As previously amended, clause a recites "requesting the object from a text and numeric centric line of business application" and clause f recites "the content manager application responding to the graphical user interface centric content manager client and rendering the object from the text and numeric centric line of business application to the graphical user interface centric user."

This moves the concept of a "method of rendering an object from a text and numeric centric line of business application to a graphical user interface centric content manager client application" from just the preamble to the body of the claim.

Applicants have further amended the claims by positively reciting:

said text and numeric centric line of business application resident on a
a resource manager, said resource manager including
(i) an HTTP server,
(ii) a store object agent,
(iii) a resource manager tracking table data base, and

(iv) *a file system;*

This amendment puts the method steps in the context of Figure 2 and paragraph [0020]

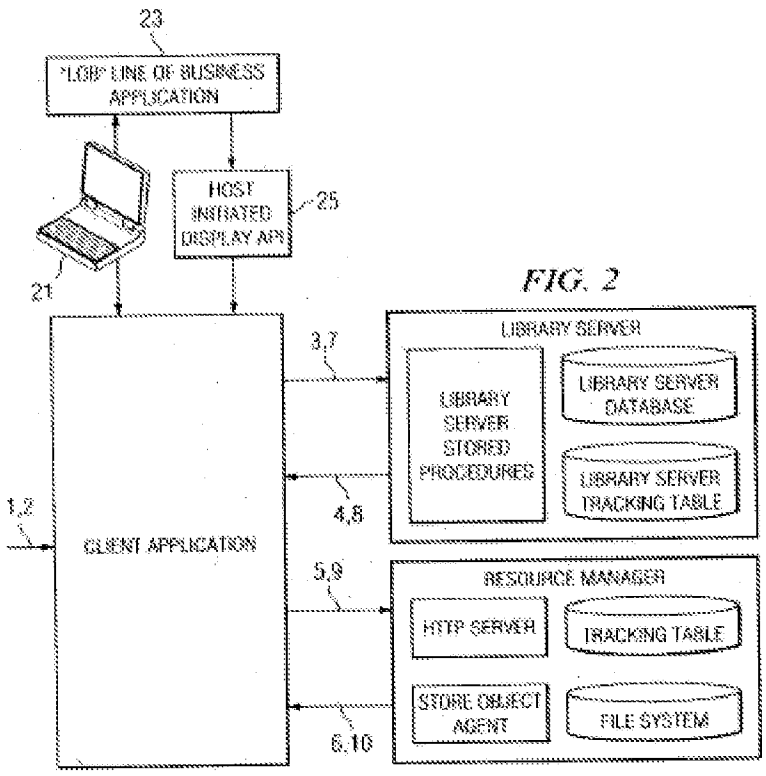

FIG. 2

and

[0020] FIG. 2 illustrates generally the client, the library server, the resource server, with the additional host initiated display application program interface between the client application and the line of business application. As shown in FIG. 2, an end user at a customer application, 21, requests an object, for example, a display, a print, folder contents, or the like, from a line of business application, 23. The line of business application initiates a "host initiated display" application program interface (`api"), 25, and calls a workstation listener, which launches the graphical user interface client, for example, the Content Manager (CM) graphical user interface (GUI) client. The line of business application host, 23, sends the customer application request to the workstation listener, 4. The CM GUI client builds a request for the object and sends the request to the content manager application, 7, for host initiated display. The content manager application responds to the content manager client and renders the object to user, 8.

Neither Christfort nor Moore teaches or suggests this system carrying out this process.

In discussing the art rejection, we turn first to Christfort. Applicants' claims recite a method of rendering an object from a text and numeric centric line of business application to a graphical user interface centric content manager client application (resident in a now positively recited computer system). The claimed method of rendering the object comprises the claimed steps of requesting the object from the text and numeric centric line of business application; with the line of business application initiating an associated host initiated display application program interface, and calling a workstation listener. Next, Applicant's claims recite a content manager host sending the customer application request to a workstation listener, and the workstation listener launching an associated content manager graphical user interface client. The content manager graphical user interface client is then claimed to build a request for the object and send the request to the associated content manager application for the associated host initiated display. Applicants' claims next recite the content manager application responding to the interface centric content manager client and rendering the text and numeric centric application object to the user. As recited in the amended claims:

> A method of rendering an object from a text and numeric centric line of business application to a graphical user interface centric content manager client application said text and numeric centric line of business application resident on a a resource manager, said resource manager including
> (i) an HTTP server,
> (ii) a store object agent,
> (iii) a resource manager tracking table data base, and
> (iv) a file system;
> said method comprising:
>
> a. requesting the object from a text and numeric centric line of business application;
>
> b. the line of business application initiating an associated host initiated display application program interface, and calling a workstation listener;
>
> c. a content manager host sending customer application request to a workstation listener;
>
> d. the workstation listener launching an associated content manager graphical user interface client;

e. the content manager graphical user interface client building a request for the object and sending the request to the associated content manager application for the associated host initiated display; and

f. the content manager application responding to the graphical user interface centric content manager client and rendering the object from the text and numeric centric line of business application to the graphical user interface centric user.

Applicant's claimed invention is materially different from Christfort's disclosure of

> "[0059] Techniques are provided for facilitating the creation and deployment of applications that are used to provide services for access by devices such as mobile clients. **These techniques include the development of applications that can be executed on a variety of devices by tailoring the output, after it has been generated by the application, based on the particular circumstances of the end user's use of the application, such as the capabilities of a mobile client or the network conditions existing at the time a customer requests service from the application.** Also, these techniques include combining the output, capabilities, and features of services together, including techniques for allowing an end user to return to a previously accessed service. In addition, these techniques include storing data at an intermediary for access by the applications associated with a service using variables and a mapping of the variables to the stored data."

The underlined clause was cited as anticipatory of "requesting an object."

Christfort further discloses that

> "[0063] Host server 110 may be implemented on one or more servers at an intermediary, such as a hosting service provider, also known as a host provider or simply as a host. The function of the host is to install and maintain applications, such as on host server 110, that are developed by either the host provider or other application developers. **The applications are typically part of a service, such as a web site, a paging service, or a telecommunications service.** The host may also provide "partial" or "shared" hosting of applications in which the applications are stored on servers associated with the application developer or service provider, but the applications may be accessed via the host. Partial or shared hosting of applications is distinguished from portal applications that are stored on servers associated with the application developer or service provider but which are not accessed via the host. End users access the services offered by other parties and companies via the host by interacting with the hosted and partially hosted applications."

This was cited for anticipating a line of business application.

Christfort, paragraph [0087] described a development process (i.e., "In one embodiment, to create a hosted application, the development website provides the developer or user with an interface for writing and editing code for the application. The interface may include an editing window or edit field that the user may use to type in the code for the

application. Similarly, to edit an existing application, the user is presented with an interface that displays the existing application code to the user in an editing window that allows the user to edit the code of the selected application.") is not Applicant's claimed "method of rendering an object from a text and numeric centric line of business application to a graphical user interface centric content manager client application" nor is it the hosted application of paragraph [0094].

Paragraph [0163] was cited for teaching sending a customer application request to a work station listener, for launching an associated content manager graphical user interface application, and building a request for an object. Paragraph [0163] describes

> "[0163] FIG. 4 is a block diagram illustrating an example of producing output using a shared hosted application, according to one embodiment of the invention. FIG. 4 illustrates a client device 410, such as a laptop computer or mobile phone, that is connected to an HTTP listener 420, such as a web server that provides web pages in response to requests. HTTP listener 420 may provide client device 410 with a web page containing a list of services associated with a hosting service 430. Upon selection of a particular service by client device 410, HTTP listener 420 sends a request for the particular service to a service linker 432 that is part of hosting service 430. Service linker 432 may be implemented on one or more servers associated with hosting service 430. Upon receipt of the request, service linker 432 identifies the service or application that is the subject of the request and forwards the request from client device 410 to a service provider 440."

While using similar words and phrases, Christford describes a materially different process then Applicants' claimed process, and there is no disclosure of Applicants' claimed "rendering an object from a text and numeric centric line of business application to a graphical user interface centric content manager client application."

Moreover, Christfort et al. describes a "style sheet" dependent solution. In this regard see numbered paragraph [0073][1], final sentence, numbered paragraphs [0097]-[0098][2],

---

[1] "In another embodiment, the application produces a comprehensive set of output that is customized or formatted by middleware transformer 112 based on a style sheet selected based on the client device. [0073], final sentence.

[2] [0097] ... According to one embodiment, prior to providing the output to a client, one or more extensible stylesheet language (XSL) style sheets are selected based on the device type of the client. XSL style sheets are discussed in more detail in U.S. application Ser. No. 09/631,884, filed Aug. 4, 2000, the entire disclosure of which is hereby incorporated by reference as if fully set forth herein. ... The application developer achieves any device support without having to do any special programming because the applications are designed to produce the same XML output regardless of the device used to request the

numbered paragraph [0099][3], and numbered paragraph [0127][4]. This passage is no longer relied upon in rejecting the claims.

Thus, Christfort et al. proposes an inapplicable solution to an inapplicable problem, and is not a proper reference.

Turning now to Moore et al., Moore et al. do not overcome the fundamental shortcomings of Christfort, with the cited provisions of Moore, column 2, lines 64-65, and column 7, lines 16-18 neither teaching nor suggesting any aspect of Applicant's claimed invention.

Moore, column 4, lines 62-65 describes

> "The stored attributes are saved in partitions in the repository 16 and organized into certain files identified as an HRD (Host Reply Definition) file 22, which is a flat file that contains reply information for each field in a screen; a REQ (or Request) file 23, which is a flat file that stores information requested by the system for each field; and, a RECOGNITION file 24, which is a flat file that associates a screen object with recognizable text. The RECOGNITION file 24 contains a list of identifiers for the screens. The HRD and REQ files are used to build type libraries that contain detailed information about a screen display."

and

> "Referring now to FIG. 4, a print of the User Interface for the present invention (i.e., the component builder). The process is begun by using the pull down menu 38."

---

service and since the device specific formatting takes place outside the applications, such as by an intermediary applying device specific XSL style sheets."

[0098] The selected style sheets are applied to the XML output from the service to produce customized output that is specifically formatted for the client that requested the service…."

[3] [0099] Further, because the execution of the code and the application of XSL style sheets to the output thereof occur via the host, all hosted and portal applications automatically gain device specific support for new devices whenever the host installs XSL style sheets for the new devices.

[4] [0127] In one embodiment, after the condition specific output has been created, an XSL style sheet may be applied to format the output according to the needs of the client to which the output is to be sent. In an alternative embodiment, in addition to the output processing described above, the middleware transformer formats the output for the specific device, either by applying one or more XSL style sheets or by any other means.

which were cited for a showing of (flat file) displays. This does not teach or suggest Applicant's claimed graphical user interface centric application, exemplified above.

Discussion of "Listener" Applicants disclose a "listener"

> [0013] As used herein a listener is a software module or application used to facilitate the association between a connection and a destination. and deployed message-driven bean. More specifically, a listener starts a service on a received command, and provides an appropriate link.

while Cristfort et al describe a "browser"

> [0004] Users of the World Wide Web use a client program, referred to as a browser, to request, decode and display information from listeners. When the user of a browser selects a link on an HTML page, the browser that is displaying the page sends a request over the Internet to the listener associated with the Universal Resource Locator (URL) specified in the link. In response to the request, the listener transmits the requested information to the browser that issued the request. The browser receives the information, presents the received information to the user, and awaits the next user request.
>
> [0005] Traditionally, the information stored on listeners is in the form of static HTML pages. Static HTML pages are created and stored at the listener prior to a request from a web browser. In response to a request, a static HTML page is merely read from storage and transmitted to the requesting browser. Currently, there is a trend to develop listeners that respond to browser requests by performing dynamic operations. For example, a listener may respond to a request by issuing a query to a database, dynamically constructing a web page containing the results of the query, and transmitting the dynamically constructed HTML page to the requesting browser.

The functions of Applicants' listener are:

1) facilitate the association between a connection and a destination. and deployed message-driven bean.
2) start a service on a received command, and
3) provide an appropriate link.

This is in contrast to Christfort's different functions of

1) issuing a query to a database,

2) dynamically constructing a web page containing the results of the query, and

3) transmitting the dynamically constructed HTML page to the requesting browser.

Discussion of "style sheet" Christfort describes a "style sheet" in the context of "middleware" [0073], providing a style sheet in the context where "prior to providing the output to a client, one or more extensible stylesheet language (XSL) style sheets are selected based on the device type of the client" [0097] which is then applied to the XML output [0098]-[0099] where the centrality of style sheets is clearly illustrated.

> [0098] The selected style sheets are applied to the XML output from the service to produce customized output that is specifically formatted for the client that requested the service. For example, the same XML output by a mobile application may ultimately be manifested in the following three ways, depending on the type of client that requested the service: (1) a list of options on the display of a mobile phone, (2) a pull down menu on a browser, and (3) an orally presented list of options on the voice interface of a telephone. The application developer achieves any device support without having to do any special programming because the applications are designed to produce the same XML output regardless of the device used to request the service and since the device specific formatting takes place outside the applications, such as by an intermediary applying device specific XSL style sheets.

> [0099] Further, because the execution of the code and the application of XSL style sheets to the output thereof occur via the host, all hosted and portal applications automatically gain device specific support for new devices whenever the host installs XSL style sheets for the new devices. The application developer need not make any changes to the application code to support any additional or newly developed client devices. Of course, if the developer wants to take advantage of improved or added capabilities, the developer may update the application to provide addition output segments or variations for such capabilities.

The centrality of style sheets is a key element of Christfort, which is not part of applicant's claimed invention.

It is submitted that the claims now pending, are properly allowable to Applicants.

Conclusion

Based on the above discussion, it is respectfully submitted that the pending claims describe an invention that is statutory subject matter and is properly allowable to the Applicants.

If any issues remain unresolved despite the present amendment, the Examiner is requested to telephone Applicants' Attorney at the telephone number shown below to arrange for a telephonic interview before issuing a Final Action.

Applicants would like to take this opportunity to thank the Examiner for a thorough and competent examination and for courtesies extended to Applicants' Attorney.

Respectfully Submitted

Certificate of Deposit

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being electronically deposited with the United States Patent and Trademark Office on the date shown below.

Date of deposit: June 3, 2007

Person mailing paper: Richard M. Goldman

Signature: /s/ Richard M. Goldman

/s/ Richard M. Goldman
Richard M. Goldman

Richard M. Goldman, Reg. # 25,585
371 Elan Village Lane, Suite 208
San Jose, CA 95134
Voice: 408-324-0716
Fax: 408-324-0672
E-mail: goldmanptn@aol.com

[i] [0059] Techniques are provided for facilitating the creation and deployment of applications that are used to provide services for access by devices such as mobile clients. These techniques include the development of applications that can be executed on a variety of devices by tailoring the output, after it has been generated by the application, based on the particular circumstances of the end user's use of the application, such as the capabilities of a mobile client or the network conditions existing at the time a customer requests service from the application. Also, these techniques include combining the output, capabilities, and features of services together, including techniques for allowing an end user to return to a previously accessed service. In addition, these techniques include storing data at an intermediary for access by the applications associated with a service using variables and a mapping of the variables to the stored data.

[ii] [0063] Host server 110 may be implemented on one or more servers at an intermediary, such as a hosting service provider, also known as a host provider or simply as a host. The function of the host is to install and maintain applications, such as on host server 110, that are developed by either the host provider or other application developers. The applications are typically part of a service, such as a web site, a paging service, or a telecommunications service. The host may also provide "partial" or "shared" hosting of applications in which the applications are stored on servers associated with the application developer or service provider, but the applications may be accessed via the host. Partial or shared hosting of applications is distinguished from portal applications that are stored on servers associated with the application developer or service provider but which are not accessed via the host. End users access the services offered by other parties and companies via the host by interacting with the hosted and partially hosted applications.

[iii] [0067] FIG. 1A also shows a connection 150 between end user 130 and user 120, and a connection 152 between end user 134 and a user 124. Connections 150 and 152 permit direct communication between the end users and the users, thus illustrating that not all communications between end users and users pass through host server 110. For example, end user 130 may be a desktop computer and therefore is able to interact directly with user 120 when requesting a service. However, the service that end user 130 requests from user 120 may be supplied by a hosted application located on host server 110. In response to the service request from end user 130, user 120 may send a request to host server 110 to provide end user 130 with output from the hosted application via connection that satisfies the request. In response to the request from user 120, host server 110 may then execute the appropriate hosted application and send the resulting output to end user 130 over connection 140.

[iv] [0087] The deployment of a hosted application may involve several steps, such as initially creating the application, subsequently editing of the application, and testing of the application. In one embodiment, to create a hosted application, the development website provides the developer or user with an interface for writing and editing code for the application. The interface may include an editing window or edit field that the user may use to type in the code for the application. Similarly, to edit an existing application, the user is presented with an interface that displays the existing application code to the user in an editing window that allows the user to edit the code of the selected application.

[v] [0091] Accorded to another embodiment, both the application code for a hosted application and the code that causes the generation of the user interface used to enter and edit the code are stored on one or more servers associated with the development provider. Consequently, the only clientside software required to develop and deploy a mobile application is a web browser, such as Netscape Navigator.

[vi] [0094] In another embodiment, to create a shared hosted application, the user writes either a portaltogo XML document or an application program that generates a portaltogo XML document as output. The terms

"partially hosted application" or the "shared hosted application" may be used to refer either to the XML document or the application that generates an XML document as output. The shared hosted application may be saved, for example, at the application developer's own website. The user then associates a URL with the shared hosted application using, for example, an HTTP listener/web server that services the application developer's web site. The shared hosted application is then added as a "service" by logging into the development website, or the SDK website, and providing the name of the service and the URL associated with the shared hosted application.

[vii] [0163] FIG. 4 is a block diagram illustrating an example of producing output using a shared hosted application, according to one embodiment of the invention. FIG. 4 illustrates a client device 410, such as a laptop computer or mobile phone, that is connected to an HTTP listener 420, such as a web server that provides web pages in response to requests. HTTP listener 420 may provide client device 410 with a web page containing a list of services associated with a hosting service 430. Upon selection of a particular service by client device 410, HTTP listener 420 sends a request for the particular service to a service linker 432 that is part of hosting service 430. Service linker 432 may be implemented on one or more servers associated with hosting service 430. Upon receipt of the request, service linker 432 identifies the service or application that is the subject of the request and forwards the request from client device 410 to a service provider 440.

[viii] [0164] Service provider 440 includes an HTTP server 442 for handling communications between service provider 440 and other servers, such as service linker 432 or servers linked together as part of the Internet. Service provider 440 also includes an application server 446 for directing requests received by HTTP server 442 to the appropriate application.

[ix] [0166] Service provider 440 also includes a database 470. For example, if application 450 is a dining directory module, database 470 may include information about restaurants in a number of cities. Upon selection of a particular city by the end user, application 450 may generate output 454 that includes a listing of restaurants in the chosen city.

[x] [0167] Service provider 440 responds to the request received from client device 410 via service linker 432 with a generic output 480. For example, generic output may be an electronic document containing portaltogo XML.

[xi] [0168] Generic output 480 is sent from service provider 440 to hosting service 430, where the generic output is processed by a device transformer 436. Device transformer 436 may be a server that functions as a middleware transformer by applying style sheets to generic XML output to a customized output 490, as described above. Customized output is then sent to client device 410 for display to the end user.

[xii] The stored attributes are saved in partitions in the repository 16 and organized into certain files identified as an HRD (Host Reply Definition) file 22, which is a flat file that contains reply information for each field in a screen; a REQ (or Request) file 23, which is a flat file that stores information requested by the system for each field; and, a RECOGNITION file 24, which is a flat file that associates a screen object with recognizable text. The RECOGNITION file 24 contains a list of identifiers for the screens. The HRD and REQ files are used to build type libraries that contain detailed information about a screen display.

[xiii] Referring now to FIG. 4, a print of the User Interface for the present invention (i.e., the component builder). The process is begun by using the pull down menu 38.